
TP-NOTE(1) Version 1.5.0 | Tp-Note documentation

Table of Contents

| | |
|---------------------------------------------------------------------|----|
| NAME | 1 |
| SYNOPSIS | 1 |
| DESCRIPTION | 2 |
| OPERATION MODES | 2 |
| New note based on clipboard data | 2 |
| New note with empty clipboard | 4 |
| New note based on a non-tp-note-file | 5 |
| Editing notes | 5 |
| Automatic filename synchronization before and after editing | 6 |
| OPTIONS | 6 |
| THE NOTE'S DOCUMENT STRUCTURE | 7 |
| META-DATA FILENAME SYNCHRONIZATION | 7 |
| CUSTOMIZATION | 9 |
| Template variables | 9 |
| Content-template conventions | 10 |
| Filename-template convention | 10 |
| Register your own external text editor for usage with Tp-Note | 11 |
| Change the markup language | 11 |
| RESOURCES | 12 |
| COPYING | 12 |
| Contribution | 12 |
| AUTHORS | 12 |

NAME

Tp-Note - fast note taking with templates and filename synchronization.

SYNOPSIS

```
tp-note [-V] [-b] [-d] [-v] [-c <config-file>] [<path>]
```

DESCRIPTION

Tp-Note is a note-taking-tool and a template system, that consistently synchronizes the note's meta-data with its filename. *Tp-Note* collects various information about its environment and the clipboard and stores them in variables. New notes are created by filling these variables in predefined and customizable *Tera*-templates. In case “<path>” points to an existing “*Tp-Note*”-file, the note's meta-data is analysed and, if necessary, its filename is modified. For all other file types, *Tp-Note* creates a new note that annotates the file “<path>” points to. If “<path>” is a directory (or, when omitted the current working directory), a new note is created in that directory. After creation, *Tp-Note* launches an external text editor of your choice. Although the note's structure follows “pandoc”-conventions, it is not tied to any specific markup language.

After the user finished editing, *Tp-Note* analyses eventual changes in the notes meta-data and renames, if necessary, the file, so that its meta-data and filename are in sync again.

OPERATION MODES

Tp-Note operates in 4 different modes, depending on its commend-line-arguments and the clipboard state. Each mode is usually associated with one content-template and one filename-template.

New note based on clipboard data

When “<path>” is a directory and the clipboard is not empty, the clipboard's content is stored in the variable “{{ clipboard }}”. In addition, if the content is a hyperlink in markdown format, the hyperlink's name is stored in “{{ clipboard_linkname }}”, and its url in “{{ clipboard_linkurl }}”. The new note is then created with the “`tmpl_clipboard_content`” and the “`tmpl_clipboard_filename`” templates. Finally, the newly created file is opened with an external text editor. When the text editor closes, *Tp-Note* synchronizes with the template “`tmpl_sync_filename`” the note's meta-data and its filename.

Note: this operation mode also empties the clipboard (configurable feature).

The clipboard contains a string

Example: While launching *Tp-Note* the clipboard contains the string: “Who Moved My Cheese?” and <path> is a directory.

```
> tp-note "./03-Favorite Readings/"
```

or

```
> cd "./03-Favorite Readings/"
> tp-note
```

This creates the document:

```
"./03-Favorite Readings/20200306-Who Moved My Cheese--Note.md"
```

with the content:

```
---
title:      "Who Moved My Cheese?"
subtitle:   "Note"
author:     "getreu"
date:       "March 6, 2020"
lang:       "en_GB.UTF-8"
revision:   "1.0"
---
```

We see from the above example, that the default template created a document with some meta-data, but without content. However, if desired or necessary it is possible to adapt the template in *Tp-Note*'s configuration file. Please note, that the filename is a simplified and sanitized concatenation of: date, title and subtitle.

The clipboard contains a markdown link

Example: `<path>` is a directory, the clipboard is not empty and it contains the string: “[The Rust Book](https://doc.rust-lang.org/book/)”.

```
> tp-note './doc/Lecture 1'
```

This creates the following document:

```
"./doc/Lecture 1/20200307-The Rust Book--Notes.md"
```

```
---
title:    "The Rust Book"
subtitle: "Notes"
author:   "getreu"
date:     "March 7, 2020"
lang:     "en_GB.UTF-8"
revision: "1.0"
---
```

```
[The Rust Book](https://doc.rust-lang.org/book/)
```

New note with empty clipboard

In case the clipboard is empty while starting, another set of templates is used to create the new note: “`tmpl_new_content`” and “`tmpl_new_filename`”. By default, the new note’s title is the parent’s directory name. The newly created file is then opened with an external text editor, allowing to change the proposed title and to add other content. When the text editor closes, *Tp-Note* synchronizes the note’s meta-data and its filename. This operation is performed with the “`tmpl_sync_filename`” template.

Example: the clipboard is empty and `<path>` is a directory (or empty):

```
> tp-note "./03-Favorite Readings/"
```

or

```
> cd "./03-Favorite Readings"
> tp-note
```

creates the document:

```
"./03-Favorite Readings/20200306-Favorite Readings--Note.md"
```

with the content:

```
---
title:    "Favorite Readings"
subtitle: "Note"
---
```

```
author:      "getreu"
date:       "March 6, 2020"
lang:      "en_GB.UTF-8"
revision:   "1.0"
---
```

New note based on a non-tp-note-file

When “<path>” points to a file whose extension is other than “.md”, a new note is created with a similar filename and a reference to the original file copied into the note. The logic of this is implemented in the templates: “`tmpl_annotate_content`” and “`tmpl_annotate_filename`”. Once the file is created, it is opened with an external text editor. After editing the file, it will be - if necessary - renamed to be in sync with the note’s meta-data.

Example:

```
> tp-note "Classic Shell Scripting.pdf"
```

creates the note:

```
"Classic Shell Scripting--Note.md"
```

with the content:

```
---
title:      "Classic Shell Scripting"
subtitle:   "Note"
author:     "getreu"
date:      "March 6, 2020"
lang:      "en_GB.UTF-8"
revision:   "1.1"
---
```

```
[Classic Shell Scripting.pdf](Classic Shell Scripting.pdf)
```

Editing notes

If not invoked with “`--batch`”, *Tp-Note* launches an external text editor after creating a new note. This also happens when “<path>” points to an existing “.md”-file.

Example: edit the note from the previous example:

```
> cd "./03-Favorite Readings"
> tp-note 20200306-Favorite Readings--Note.md
```

Automatic filename synchronization before and after editing

Before launching the text editor and after closing it, *Tp-Note* synchronizes the filename with the note's metadata. When the user changes the metadata of a note, *Tp-Note* will replicate that change in the note's filename. As a result, *all your note's filenames always correspond to their metadata*, which allows you to find your notes back quickly.

Example:

```
> tp-note "20200306-Favorite Readings--Note.md"
```

Tp-Note opens the file in a text editor. Now the note-taker decides to update the title in the note's YAML metadata section from `title: "Favorite Readings"` to `title: "Introduction to bookkeeping"`. After closing the text editor the filename is automatically updated to:

```
"20200306-Introduction to bookkeeping--Note.md"
```

Note: the sort-tag `"20200306"` has not changed. The filename synchronization mechanism never does. (See below for more details about filename synchronization).

OPTIONS

-b, --batch

Create a new file or rename the file to stay synchronized with its meta-data, but does not launch the external text editor.

-c CF, --config=CF

Load the alternative config file *CF* instead of the default one.

-d, --debug

Print additional log-messages on console. It shows the available template variables, the templates used and the rendered result of the substitution. This option particularly useful for

debugging new templates. On Windows, the output must be redirected into a file to see it. To do so open the command-prompt and type:

```
tp-note.exe -d >debug.txt 2>&1
```

-v, --view

Launch the external text editor, if possible, in read-only-mode.

-V, --version

Print *Tp-Note*'s version and exit.

THE NOTE'S DOCUMENT STRUCTURE

A *Tp-Note*-note file is always UTF-8 encoded. As newline, either the Unix standard “`\n`” or the Windows standard “`\r\n`” is accepted. *Tp-Note* writes out newlines according the operating system it runs on.

Tp-Note is designed to be compatible with “`Pandoc`’s and `Markdowns` document structure as shown in the figure below.

```
-----  
---  
<YAML-front matter>  
---  
<document-body>  
-----
```

The YAML front matter starts at the beginning of the document with “`---`” and ends with “`...`” or “`---`”. Note that according to the YAML standard, string-literals are always encoded as JSON strings.

There is no restriction about the markup language used in the note’s text body. However, the default templates assume that markdown and the file extension “`.md`” is used. Both can be changed easily by adapting *Tp-Note*’s configuration file.

META-DATA FILENAME SYNCHRONIZATION

Consider the following *Tp-Note*-file:

```
-----  
20151208-Make this world a better place--Suggestions.md  
-----
```

The filename has 4 parts:

.....

```
{{ sort-tag }}-{{ title }}--{{ subtitle }}.{{ extension }}
```

.....

A so called *sort-tag* is a numerical prefix at the beginning of the filename. It is used to order files and notes in the file system. Besides numerical digits, a *sort-tag* can be any combination of 0123456789-_¹ and is usually used as

- *chronological sort-tag*

.....

```
20140211-Reminder.doc
20151208-Manual.pdf
```

.....

- or as a *sequence number sort-tag*.

.....

```
02-Invoices
08-Tax documents
09_02-Notes
```

.....

When *Tp-Note* creates a new note, it prepends automatically a *chronological sort-tag* of today. The “`{{ title }}`” part is usually derived from the parent directory name omitting its own *sort-tag*.

A note’s filename is in sync with its meta-data, when the following is true (slightly simplified, see the configuration file for the complete definition):

```
filename on disk without sort-tag == “-{{ title }}--
{{ subtitle }}.md”2
```

Consider the following document with the filename:

.....

```
20200306-My file.md
```

.....

and the content:

.....

```
---
title:      "1. The Beginning"
subtitle:   "Note"
```

¹ The characters “_” and “-” are not considered to be part of the *sort-tag* when they appear in first or last position.

² The variables “`{{ title }}`” and “`{{ subtitle }}`” reflect the values in the note’s metadata.


```
author:      "getreu"  
date:       "March 6, 2020"  
lang:       "en_GB.UTF-8"  
revision:   "1.1"  
---
```

As “-My file.md” is not equal to “- '1. The Beginning--Note.md”, *Tp-Note* will rename the file to “20200306- '1. The Beginning--Note.md”.

If the filename had been “05_02-My file.md”, it would rename it to “05_02- '1. The Beginning--Note.md”.

Note: *Tp-Note* never changes a sort-tag, but might change the rest of the filename!

CUSTOMIZATION

Tp-Note's configuration file resides typically in “~/ .config/tp-note/tp-note.toml” on Unix or in “C:\Users\<LOGIN>\AppData\Roaming\tp-note\config\tp-note.toml>” on Windows. When *Tp-Note* starts, it tries to find its configuration file. If it fails, it writes a default configuration file. *Tp-Note* is best customized by starting it once, and then modifying its default configuration.

The configuration file is encoded according to the TOML-standard. Variables starting with “`tmpl_*`” are *Tera-Template*-strings (see: <https://tera.netlify.com/docs/#templates>).

Tp-Note captures and stores its environment in *Tera-variables*. For example, the variable “`{{ dirname }}`” is initialized with the document's parent directory. The variable “`{{ clipboard }}`” contains the content of the clipboard. To learn more about variables, launch *Tp-Note* with the “`--debug`” option and observe what information it captures from its environment.

Template variables

All [Tera template variables and functions](https://tera.netlify.com/docs/#templates)³ can be used within *Tp-Note*. For example “`{{ get_env(name='LANG') }}`” gives you access to the “LANG” environment variable.

In addition, *Tp-Note* defines the following variables:

- “`{{ sort_tag }}`”: the sort-tag of the current note, e.g. “01-23” or “20191022”,
- “`{{ dirname }}`”: the parent directory's name,

³ <https://tera.netlify.com/docs/#templates>

- “`{{ file_stem }}`”: the note’s filename without extension,
- “`{{ clipboard }}`”: all the text from the clipboard,
- “`{{ clipboard_truncated }}`”: the first 200 bytes from the clipboard,
- “`{{ clipboard_heading }}`”: the clipboard’s content until end of the first sentence, or the first newline.
- “`{{ clipboard_linkname }}`”: the name of the first Markdown formatted link in the clipboard,
- “`{{ clipboard_linkurl }}`”: the URL of the first Markdown formatted link in the clipboard,
- “`{{ extension }}`”: the filename extension of the existing note on disk,
- “`{{ note_extension }}`”: the default extension for new notes (which can be changed in the configuration file),
- “`{{ username }}`”: the content of the first non-empty environment variable: `LOGNAME`, `USER` or `USERNAME`.
- “`{{ title }}`”: the title as indicated in the YAML front matter of the note (only available in filename-templates).
- “`{{ subtitle }}`”: the subtitle as indicated in the YAML front matter of the note (only available in filename-templates).

Content-template conventions

Tp-Note distinguishes two template types: content-templates “`tmpl*_content`” used to create the note’s content (front-matter and body) and filename-templates “`tmpl*_filename`” used to calculate the note’s filename.

Strings in the YAML front matter of content-templates are JSON encoded. Therefore all variables used in the front matter must pass an additional “`json_encode()`”-filter. For example, the variable “`{{ dirname }}`” becomes “`{{ dirname | json_encode() }}`” or just “`{{ dirname | json_encode }}`”.

Filename-template convention

The same applies to filename-template-variables: in this context we must guarantee, that the variable contains only file system friendly characters. For this purpose *Tp-Note* provides the additional Tera filters “`path`” and “`path(alpha=true)`”.

- The “`path()`” filter transforms a string in a file system friendly form. This is done by replacing forbidden characters like “`?`” and “`\`” with “`_`” or space. This filter can be used

with any variables, but is most useful with filename-templates. For example, take a look at the “`tmpl_sync_filename`” template: it starts with “`{{ sort_tag | path }}`”.

- “`path(alpha=true)`” is similar to the above, with one exception: when a string starts with a digit “0-9”, the whole string is prepended with ‘`.`’. For example: “`1 The Show Begins`” becomes “`.1 The Show Begins`”. This filter should always be applied to the first variable assembling the new filename, e.g. “`{{ title | path(alpha=true) }}`”. This way, it is always possible to distinguish the sort-tag from the actual filename.

In filename-templates all variables must pass either the “`path`” or the “`path(alpha=true)`” filter!

Register your own external text editor for usage with Tp-Note

The configuration file variables “`editor_args`” and “`viewer_args`” define a list of external text editors to be launched for editing. “`viewer_args`” is used when *Tp-Note* is invoked with “`--view`” in viewer mode. The list contains well-known text editor names and its command-line arguments. *Tp-Note* tries to launch every text editor from the beginning of the list until it finds an installed text editor.

In order to use your own text editor, just place it at the top of the list. To make this work properly, make sure, that your text editor does not fork! You can check this when you launch the text editor from the command-line: if the prompt returns immediately, then it forks the process. In contrast, it is Ok when the prompt only comes back at the moment when the text editor is closed. Many text editors provide an option not to fork: for example the *VScode*-editor can be launched with the “`--wait`” option and `vim` with `vim --nofork`. However, *Tp-Note* also works with forking text editors. Then, the only drawback is, that *Tp-Note* can not synchronize the filename with the note’s metadata when the user has finished editing. It will still happen, but only when the user opens the note again with *Tp-Note*.

Change the markup language

Tp-Note is markup language agnostic, however the default templates define *Markdown* as default markup language. To change this, just edit the following 2 templates:

1. Change the variable “`note_extension='md'`” to e.g. “`note_extension='rst'`”
2. The last line in template “`tmpl_clipboard_content`” defines a hyperlink in Markdown format. Change the link format according to your markup language convention.

RESOURCES

Tp-Note is hosted on:

- Github: <https://github.com/getreu/tp-note> and on
- Gitlab (mirror): <https://gitlab.com/getreu/tp-note>.

COPYING

Copyright (C) 2016-2020 Jens Getreu

Licensed under either of

- Apache Licence, Version 2.0 (LICENSE-APACHE or <http://www.apache.org/licenses/LICENSE-2.0>)
- MIT licence (LICENSE-MIT or <http://opensource.org/licenses/MIT>)

at your option.

Contribution

Unless you explicitly state otherwise, any contribution intentionally submitted for inclusion in the work by you, as defined in the Apache-2.0 licence, shall be dual licensed as above, without any additional terms or conditions. Licensed under the Apache Licence, Version 2.0 (the "Licence"); you may not use this file except in compliance with the Licence.

AUTHORS

Jens Getreu <getreu@web.de>